***Listing of the Claims*:**

1. (Currently Amended) A system for decoupling commercial-off-the-shelf software applications from data stores, the system comprising:

a plurality of commercial-off-the-shelf software applications each compatible with one of a plurality of first data stores, each of the plurality of commercial-off-the-shelf software applications submits a data request compatible with one of the plurality of first data stores;

a plurality of second data stores;

a plurality of drivers, wherein each of the plurality of first data stores and the plurality of second data stores has a corresponding one of the plurality of drivers configured to receive the data request and pass the data request to the corresponding data store;

~~at least one processor;~~

one of a plurality of listeners, recorded on a computer readable medium, when executed by at least one processor, simulates a corresponding one of the plurality of drivers corresponding with one of the plurality of first data stores and receives the data request from a corresponding one of the plurality of commercial-off-the-shelf software applications that is compatible with the one of the plurality of first data stores simulated by the one of the plurality of listeners, wherein each of the plurality of commercial-off-the-shelf software applications has a corresponding one of the plurality of listeners;

a translator, recorded on a computer readable medium, in communication with the one of the plurality of listeners and the plurality of second data stores, the translator, when executed by at least one processor, receives the data request from the one of the plurality of listeners, translates the data request into a generic format to produce a translated data request, and submits the translated data request for one of the plurality of drivers corresponding with one of the plurality of second data stores for storage by the one of the plurality of second data stores[[.]]; and

a data access layer, recorded on a computer readable medium, in communication with the translator and, when executed by at least one processor, determines to direct the translated data request from one of the commercial-off-the-shelf software applications that correspond with the one of the plurality of listeners to the one of the plurality of second data stores, and translates the translated data request from the generic format into a storage format of the one of the plurality of second data stores.

2. (Cancelled)

3. (Currently Amended) The system of Claim 1[[2]], wherein the data access layer maintains an enterprise data model including a data map of where to direct [[the]] a data request of each of the commercial-off-the-shelf software applications.

4. (Currently Amended) The system of Claim 3, wherein the data access layer receives the translated data request from the translator and directs the translated data request to the one of the plurality of second data stores.

5. (Currently Amended) The system of Claim 1[[2]], wherein a first commercial-off-the-shelf software application of the plurality of commercial-off-the-shelf software applications submits a first data request in a first relational database format and wherein the data access layer translates the first data request to a second relational database format.

6. (Previously Presented) The system of Claim 5, wherein a second commercial-off-the-shelf software application of the plurality of commercial-off-the-shelf software applications submits a second data request in an older version of the first relational database format and wherein the data access layer translates the second data request to a newer version of the first relational database format.

7. (Currently Amended) The system of Claim 1[[2]], wherein a first commercial-off-the-shelf software application of the plurality of commercial-off-the-shelf software applications submits a first data request in an older version of a first relational database format and wherein the data access layer translates the first data request to a newer version of the first relational database format.

8. (Currently Amended) The system of Claim 1, wherein at least one of the plurality of second data stores corresponds with one of the plurality of first data stores.

9. (Currently Amended) The system of Claim 8, wherein the at least one of the plurality of second data stores is further defined as a newer version data store of one of the plurality of first data stores.

10. (Currently Amended) The system of Claim 9, wherein at least one of the plurality of second data stores is further defined as a newer version of a relational database of a first vendor and wherein one of the plurality of first data stores is further defined as an older version of the relational database of the first vendor.

11. (Currently Amended) The system of Claim 9, wherein at least one of the plurality of second data stores is further defined as a newer version of a relational database of a second vendor and wherein one of the plurality of first data stores is further defined as an older version of the relational database of the second vendor.

12. (Previously Presented) The system of Claim 1, wherein the plurality of commercial-off-the-shelf software applications are each operable with only one of a plurality of data stores, each of the plurality of commercial-off-the-shelf software applications submitting data requests compatible with only one of the plurality of data stores.

13. (Currently Amended) A system for maintaining compatibility of commercial-off-the-shelf software applications with data stores, the system comprising:

one of a plurality of commercial-off-the-shelf software applications operable with only one of a plurality of first data stores, the one of the plurality of commercial-off-the-shelf software applications submits a data request compatible with only the one of the plurality of first data stores;

a first of a plurality of drivers configured to receive the data request and pass the data request to the one of the plurality of first data stores;

~~at least one processor;~~

one of a plurality of listeners, recorded on a computer readable medium, when executed by at least one processor, simulates the first of the plurality of drivers and receives the data request from the one of the plurality of commercial-off-the-shelf software applications submitted for the first of the plurality of drivers, wherein each of the plurality of commercial-off-the-shelf software applications has a corresponding one of the plurality of listeners;

a translator, recorded on the computer readable medium, in communication with the one of the plurality of listeners, when executed by at least one processor, receives the data request from the one of the plurality of listeners and translates the data request into a generic format to produce a first translated data request;

a data access layer, recorded on the computer readable medium, in communication with the translator and, when executed by at least one processor, determines, based on an enterprise data model, to direct the

data request of the one of the plurality of commercial-off-the-shelf software applications to one of a plurality of second data stores and translates the first translated data request from the generic format into a storage format of the one of the plurality of second data stores to produce a second translated data request;

a wrapper, recorded on the computer readable medium, when executed by at least one processor, receives the second translated data request from the data access layer and wraps the second translated data request based on the storage format of the one of the plurality of second data stores;

a second driver configured to receive the wrapped second translated data request and pass the wrapped second translated data request to the one of the plurality of second data stores; and

the one of the plurality of second data stores receives the wrapped second translated data request from the second driver and performs an action specified in the data request.

14. (Currently Amended) The system of Claim 13, wherein the one of the plurality of second data stores is one of a newer version data store of the one of the plurality of first data stores and a different vendor database than the one of the plurality of first data stores .

15. (Cancelled)

16. (Currently Amended) A system for integration of commercial-off-the-shelf software applications and databases, the system comprising:

a plurality of commercial-off-the-shelf software applications each compatible ~~operable~~ with one of a plurality of first data stores, one of the plurality of the commercial-off-the-shelf software applications submits a data request compatible with one of the plurality of first data stores;

a first of a plurality of drivers configured to receive the data request and pass the data request to the one of the plurality of first data stores;

at least one processor;

one of a plurality of listeners, recorded on a computer readable medium, when executed by at least one processor, simulates the first of the plurality of drivers and receives the data request from a corresponding one of the plurality of the commercial-off-the-shelf software applications submitted for the first of the plurality of drivers, wherein each of the plurality of commercial-off-the-shelf applications has a corresponding one of the plurality of listeners;

a translator, recorded on a computer readable medium, in communication with the one of the plurality of listeners, when executed by at least one processor, receives the data request from the one of the plurality of listeners and translates the data request into a generic format to produce a translated data request;

a data access layer, recorded on a computer readable medium, in communication with the translator and, when executed by at least one

processor, determines, based on an enterprise data model, to direct the translated data request from the one of the plurality of commercial-off-the-shelf software applications that corresponds with the one of the plurality of listeners to one of a plurality of second data stores, and translates the translated data request from the generic format into a storage format of the one of the plurality of second data stores;

a second of the plurality of drivers configured to receive the translated data request and pass the translated data request to [[a]] the one of the plurality of second data stores,[[;]] wherein the one of the plurality of second data stores receives the translated data request from the second of the plurality of drivers and performs an action specified in the data request; and

a service broker, recorded on the computer readable medium, when executed by at least one processor, maintains a record of data requests from the one of the plurality of commercial-off-the-shelf software applications and stored in the one of the plurality of second data stores, the service broker further configured to roll-back failed data requests.

17. (Cancelled)

18.(Currently Amended) The system of Claim 16, wherein the one of the plurality of commercial-off-the-shelf software applications is operable with only the one of the plurality of first data stores, and wherein the one of the plurality of commercial-off-the-shelf software applications submits the data request compatible with only the first data store.

19. (Currently Amended) The system of Claim 16, wherein the service broker further comprises:

a transaction data store configured that maintains a record of the data request by the one of the plurality of commercial-off-the-shelf software applications;

an exception handler that identifies a failed transaction and communicates with the transaction data store to restore the one of the plurality of second data stores to a state prior to the failed transaction.

20. (Currently Amended) The system of Claim 19, further comprising a data warehouse, recorded on the computer readable medium, and wherein the data warehouse, when executed by at least one processor, is asynchronously updated with the data request from the one of the plurality of commercial-off-the-shelf software applications.

21. (Original) The system of Claim 19, wherein a compensating transaction is used to restore the failed transaction.

22. (Original) The system of Claim 21, wherein an XA transaction is used in combination with the compensating transaction to restore the failed transaction.

23. (Currently Amended) The system of Claim 19, further comprising:

a data warehouse, recorded on the computer readable medium, when executed by at least one processor, maintains data;

a query processor, recorded on the computer readable medium, when executed by at least one processor, manages transaction processing of data requests from the one of the plurality of commercial-off-the-shelf software applications; and

a metadata repository, recorded on the computer readable medium, when executed by at least one processor, maintains a logical data model related to the data, wherein the metadata repository instructs the query processor regarding handling of the data requests from the one of the plurality of commercial-off-the-shelf software applications and between the one of the plurality of second data stores and the data warehouse.